

# PC Based DC Motor Speed Control using PID for Laboratory

Kyaw Zin Latt, Than Htike Aung, Zaw Min Min Htun

Department of Electronic Engineering Mandalay, Technological University, Mandalay, Myanmar

**How to cite this paper:** Kyaw Zin Latt | Than Htike Aung | Zaw Min Min Htun "PC Based DC Motor Speed Control using PID for Laboratory"

Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.2398-2400, <https://doi.org/10.31142/ijtsrd27898>



IJTSRD27898

Copyright © 2019 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



[1] The speed of the DC motor is controlled by using pulse width modulation (PWM), a technique of rapidly pulsing the power on off. Arduino microcontroller is used to receive the data, revolution per minutes, proportional gain, integral gain and derivative gain, from computer. Proportional-Integral-Derivative (PID) controllers have been widely used for speed and position control of DC motor [3]. The objective of this paper is to show that by employing the PID gains tuning system from graphical user interface (GUI) to get the optimization can be achieved [5]. The students can tune the PID gains from GUI. This GUI was developed by C# windows form application [4].

## II. Operation Principle

### A. Operation Principle

The overall block diagram is shown in Figure 1. The main objective of this work is to control DC motor using PID control algorithm and tune the PID gains from computer for laboratory. This training board consists of four components, computer, arduino, L298N hybrid motor driver and namiki DC motor. To control the speed of DC motor, student can start the motor by using computer and monitor the speed of motor from C# windows form application. And then, students can tune the PID gains from this form and test the response of speed of the DC motor. PID control algorithm is written in arduino Uno microcontroller and drive the DC motor by using L298N motor driver. From Graphical User Interface in computer set the gains and revolution per minute shown in Figure 2.

## ABSTRACT

In this paper, PC based DC motor speed control using PID (proportional, Integral, derivative) control algorithm for laboratory. Arduino Uno microcontroller and L298N hybrid motor driver module are used to drive the dc motor. PID algorithm is uploaded to arduino and proportional gain (kp), Integral gain (ki), and derivative gain ( kd) can be set and tuned via windows form application in computer. This application is developed by using c# programming. The desired speed and PID gains are set by computer and drive the DC motor. The speed of the motor can retrieve by using encoder which is included in DC motor. By using this work, student can learn how to control the speed of DC motor using PID controller and how to tune the gains.

**KEYWORDS:** PID, C#, arduino uno, L298n, windows form application

## I. INTRODUCTION

Nowadays, DC motors have been widely used in industry even though its maintenance costs are higher than the induction motor. A **DC motor** is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronics, to periodically change the direction of current flow in part of the motor.

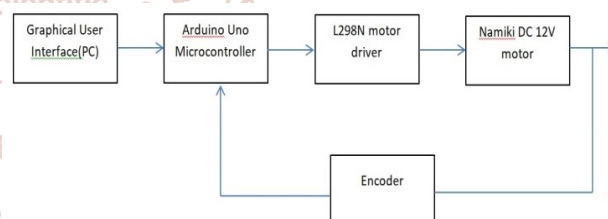


Figure1. Overall design of PC based DC speed control

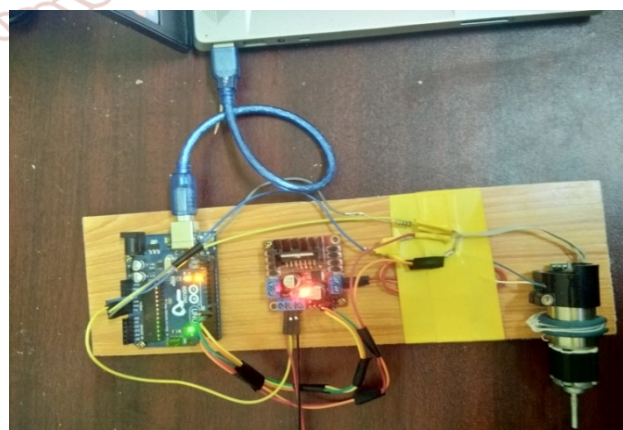


Figure2. Experiment for DC motor speed control

## III. PROPORTIONAL-INTEGRAL-DERIVATIVE CONTROLLER

### A. Proportional response

The proportional component depends only on the difference between the set point and the process variable. This difference is referred to as the Error term. The *proportional gain* determines the ratio of output response to the error

signal. For instance, if the error term has a magnitude of 10, a proportional gain of 5 would produce a proportional response of 50. In general, increasing the proportional gain will increase the speed of the control system response. However, if the proportional gain is too large, the process variable will begin to oscillate. In contrast, a small gain results in a small output response to a large input error.

$$P_{out} = k_p e(t)$$

### B. Integral Response

The integral component sums the error term over time. The result is that even a small error term will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero, so the effect is to drive the Steady-State error to zero. Steady-State error is the final difference between the process variable and set point. A phenomenon called integral windup results when integral action saturates a controller without the controller driving the error signal toward zero.

$$I_{out} = \int_0^t e(t) dt$$

### C. Derivative Response

The derivative component causes the output to decrease if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable. Increasing the *derivative time* parameter will cause the control system to react more strongly to changes in the error term and will increase the speed of the overall control system response. Most practical control systems use very small derivative time, because the derivative response is highly sensitive to noise in the process variable signal. If the sensor feedback signal is noisy or if the control loop rate is too slow, the derivative response can make the control system unstable. The concept of PID controller is depicted in Figure 3.

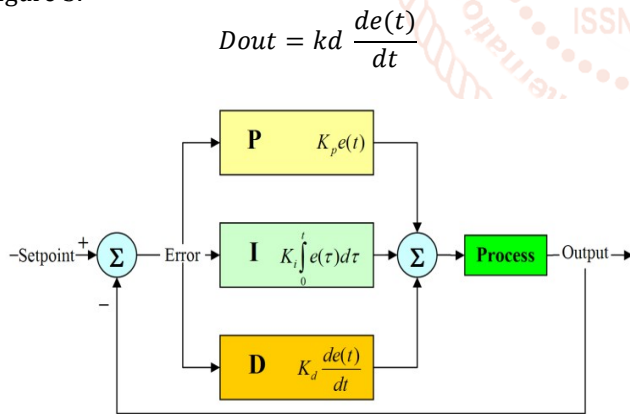


Figure3. PID controller

## IV. HARDWARE AND SOFTWARE COMPONENTS

### A. Arduino Uno

Arduino is a microcontroller and open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board and a piece of software component, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board you can simply use a USB

cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

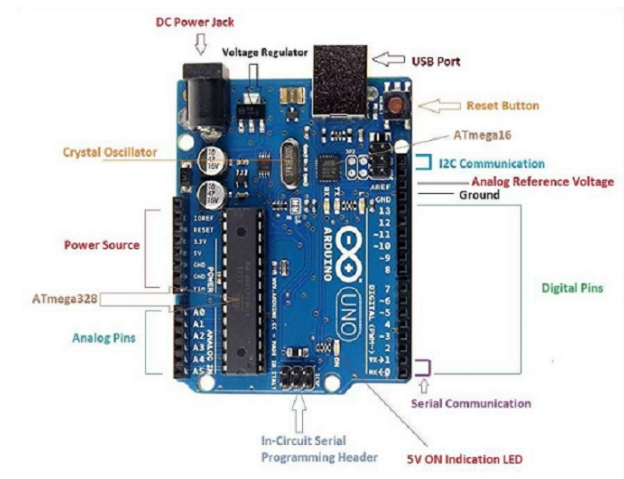


Figure6. Arduino Uno microcontroller

### B. L298N Dual H-Bridge Motor Driver

This dual bidirectional motor driver, Figure 4, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow to easily and independently control two motors of up to 2A each in both directions. It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board is equipped with power LED indicators, on-board +5V regulator and protection diodes.

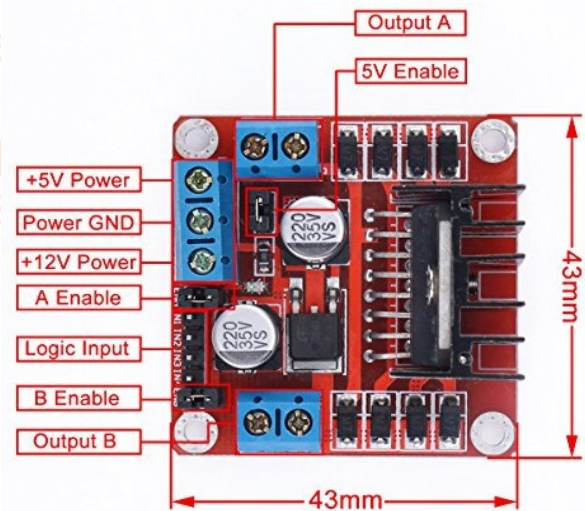


Figure4. L298N hybrid motor driver

### C. Namiki precision motor

The 12V Namiki DC gear-motor shown in Fig. 5 is a powerful motor to drive the position control system. It comes with the photoelectric encoder output and planetary gear ratio reduced by 80:1 gear. It can provide 120 rpm with the rated voltage of 12VDC [3]. To read the count values from the encoder, the user would check the condition of channel A and B rotation applying by experiment. For the rotor shaft count per revolution values, it is very important to multiply the gear ratio by count values. The specification of the motor is shown in Table 1.





Figure5. Namiki DC motor with encoder

Table1. Specification of namiki motor

Brand:	Namiki Coreless Motor (Japan)
Model:	22CL-3501PG
Operating Voltage:	12VDC
Rated power:	15W
Stall torque:	16.5kg*cm
Continuous torque:	5kg*cm
Diameter:	22mm
Shaft length:	19mm (with 90 degree double-cut)
Length:	65mm (includes encoder gear box)
Stall current:	1.8A
Reduction ratio:	80:1(metal planetary reduction gear)
Output speed:	120 RPM@12VDC

#### D. C# programming Language

C# is a hybrid of C and C++, it is a Microsoft programming language developed to compete with Sun's java language. C# programming language, windows form applications, web applications and network application can be developed. In this work, windows form application was developed. C # is an object-oriented programming language used with web service on the .Net platform and designed for development of web application.

C# boasts type-safety, interoperability, garbage collection, simplified type declarations, versioning and scalability support, and other features that make developing solutions faster and easier, especially for COM+ and Web services. Microsoft critics have pointed to the similarities between C# and Java.

#### V. RESULTS

Experiment for PC based DC motor speed control is shown in Figure 6. Desired revolution per minutes and respective PID gains send to microcontroller and drive the motor. The actual revolution per minutes of DC motor can be calculated by using encoder pulses. This RPM is sent from

microcontroller to PC via universal serial bus cable and display in GUI. Figure 6 shows the comparison of desired rpm and actual rpm graph. According to this figure, the speed of the DC motor with encoder can be control via computer by employing the gains of PID.

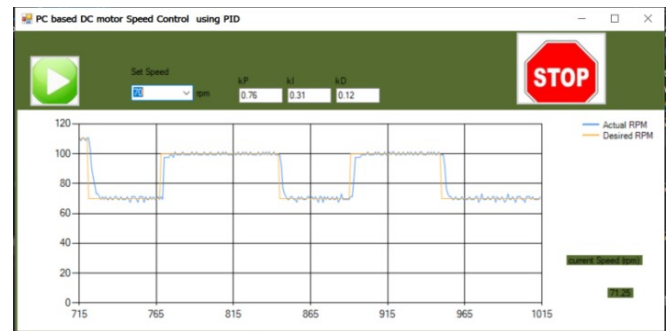


Figure6. GUI for PC based DC motor speed control using PID

#### VI. CONCLUSION

PC based DC motor speed control using PID control algorithm was designed and implemented for laboratory. Student can control the speed of DC encoder motor and tune the PID gains,  $k_p$ ,  $k_i$ ,  $k_d$  from C# the windows form application. And then, the output RPM of DC motor can be monitored from PC. So, this work is effective and efficient for the students. Teacher also can explain the PID algorithm using this training board.

#### ACKNOWLEDGMENT

I would like to express my gratitude to all the teachers from the department of Electronic Engineering, Mandalay Technological University for their suggestions.

#### VII. REFERENCES

- [1] Liu Jiang-yi1, a, Sun Shu-ying2, "Design and Implementation of Servo Motor Speed Control System Based on STM32", Applied Mechanics and Materials Vols. 427-429 (2013) pp 999-1002, 2013
- [2] Dinesh Bista, "Understanding and Design of an Arduino-based PID Controller", Virginia Commonwealth University, 2016
- [3] <https://www.aliexpress.com/item/32813911637.html>
- [4] <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>
- [5] Myo Maung Maung\*, Maung Maung Latt\*\*, Chaw Myat Nwe\*\*, "DC Motor Angular Position Control using PID Controller with Friction Compensation", International Journal of Scientific and Research Publications, Volume 8, Issue 11, 149 ISSN 2250-3153, November 2018